Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-20. (canceled)

21. (currently amended) Processing procedure for an electronic system subject to transient error constraints, comprising:

multiplexing in time a first and second virtual channels located on a single physical channel for each ~~in which in a given~~ real time cycle, ~~in other words in a given~~ said real time cycle including an operational cycle of a software task that is executed periodically and continuously, ~~two virtual sequences located on a single physical sequence are multiplexed in time (~~

storing the data resulting from each execution of a virtual ~~sequence~~ channel ~~being stored so that they can be voted before use), and in which if an error is detected~~,

voting these resulting data before using them when the two virtual channels are completed, in order to be able to detect the presence of an error,

canceling the real time cycle in progress ~~is inhibited and~~ in case an error is detected,

reloading a healthy context issued from the previous cycle in case an error is detected,

restarting in case an error is detected, that consists of executing the ~~is reloaded to make a restart that consists of a~~ nominal ~~execution of the~~ next cycle starting from the reloaded context.

22. (currently amended) Process according to claim ~~1~~ 21, in which three error confinement areas (time, software and hardware) are used.

23. (currently amended) Process according to claim ~~1~~ 21, in which a memory plane in the control unit is used, protected from singular events by an error detection and correction code.

24. (currently amended) Process according to claim ~~1~~ 21, in which the detection/correction granularity used is the real time cycle for the software tasks being performed on the computer.

25. (currently amended) Process according to claim ~~1~~ 21, in which the "backup context" function activated regularly is achieved by means of an index change.

26. (currently amended) Process according to claim ~~1~~ 21, in which the "restore context" function activated during an error correction is performed due to the fact that the index indicating the context considered to be ~~healthy, in other words~~ error free, after the previous operational cycle has not changed, even though ~~is~~ it has usually swapped, in

other words no errors are detected; this "no swap" being inherent to inhibition of the real time cycle in which the error is detected.

27. (currently amended) Process according to claim ~~1~~ 21, in which segmentation of the memory is associated with a hardware device to check access rights.

28. (currently amended) Process according to claim ~~7~~ 27, in which the hardware device to check access rights enables several access configurations, each configuration allowing access to one or several non-contiguous segments.

29. (currently amended) Process according to claim ~~7~~ 27, in which the hardware device to check access rights is used to select several access configurations with logical combinations of one or several keys.

30. (currently amended) Process according to claim ~~1~~ 21, in which the ~~variables/~~data to be voted are put into a table.

31. (currently amended) Process according to claim ~~1~~ 21, in which a software vote is used for which integrity is achieved by software checks, particularly including a software and hardware monitoring processor.

32. (currently amended) Process according to claim ~~1~~ 21, ~~in which a transfer to the control electronics is controlled by~~ wherein a hardware device ~~that~~ checks access rights

and limits the validity of this transfer in time, thus delimiting a hardware error confinement area.

33. (currently amended) Process according to claim ~~1~~ 21, used in space applications.

34. (currently amended) ~~Device~~ System for monitoring memory accesses in a computer comprising:

~~a control unit built around a microprocessor and~~ a memory, ~~in which~~ the memory ~~is~~ being partitioned into segments, ~~in which~~ each segment ~~has~~ having an access right,

a device in which registers allow to store access keys and all or some of the keys available in the device being combined using a logical function to define access right to each segment, these access rights ~~defined by a logical function of all or some of the keys available in the device, the access right to each segment~~ being checked in real time, and in which access for some segments will only be authorized if there is a very strong probability that the microprocessor will be in a good operating state, thus enabling safe storage of critical data,

wherein said access keys comprises a task number key, a virtual channel number key, and a vote key.

35. (currently amended) Device according to claim ~~14~~ 34, in which a set of non-contiguous segments is accessible, in read only for some segments and in read/write for other segments, depending on the programming of the keys present in the device.

36. (currently amended) Device according to claim ~~14~~ 34, in which the segment size is arbitrary, so that it can be optimized for a given application.

37. (currently amended) Device according to claim ~~14~~ 34, in which definitions of the set of available keys, the logical combination functions for these keys and the configuration of the accessible segments as a function of the programming of the keys, are specific.

38. (currently amended) Device according to claim ~~14~~ 34, in which one of the segments has a write authorization accessible in an exceptional state of the computer, thus enabling safe storage of critical data.

39. (currently amended) Device according to claim ~~14~~ 34, in which segments enabling safe storage of critical data are ~~stored~~ grouped by pair, working in flip-flop.